

# Simulation Study of Learning Automata Games in Automated Highway Systems<sup>1</sup>

Cem Ünsal

Postdoctoral Fellow

Robotics Institute, Carnegie Mellon University  
5000 Forbes Avenue, Pittsburgh, PA 15213-3890  
(412) 268-5594 unsal@ri.cmu.edu

Pushkin Kachroo

Assistant Professor

The Bradley Department of Electrical Engineering

Virginia Polytechnic Institute and State University

Blacksburg, VA 24062-0111

pushkin@ctr.vt.edu bay@vt.edu

John S. Bay

Associate Professor

## *Abstract*

One of the most important issues in Automated Highway System (AHS) deployment is intelligent vehicle control. While the technology to safely maneuver vehicles exists, the problem of making intelligent decisions to improve a single vehicle's travel time and safety while optimizing the overall traffic flow is still a stumbling block. We propose an artificial intelligence technique called *stochastic learning automata* to design an intelligent vehicle path controller. Using the information obtained by on-board sensors and local communication modules, two automata are capable of learning the best possible (lateral and longitudinal) actions to avoid collisions. This learning method is capable of adapting to the automata environment resulting from an unmodeled physical environment.

Although the learning approach taken is capable of providing a safe decision, optimization of the overall traffic flow is required. This is achieved by studying the interaction of the vehicles. The design of the adaptive vehicle path planner based on local information is extended to the interaction of multiple intelligent vehicles. By analyzing the situations consisting of conflicting desired vehicle paths, we extend our design by additional decision structures. The analysis of the situations and the design of the additional structures are made possible by treatment of the interacting reward-penalty mechanisms in individual vehicles as automata games. The definition of the physical environment of a vehicle as a series of discrete state transitions associated with a "stationary automata environment" is the key to this analysis and to the design of the intelligent vehicle path controller.

---

<sup>1</sup> This material is based upon work supported in part by the Naval Research Laboratory under Grant no. N00014-93-1-G022, and in part by the Center for Transportation Research and Virginia Department of Transportation under Smart Road Project.

A condensed version of this paper is presented in the *1<sup>st</sup> IEEE Conference on Intelligent Transportation Systems (ITSC'97)*, Boston, Massachusetts, Nov. 9-12, 1997.

## 1. Introduction

One of today's most serious social, economical, and environmental problems is the traffic congestion. Considering the potential of the application of intelligent systems to surface transportation problem, US Department of Transportation deems the investigation of new technologies as crucial. The technology needed to create an intelligent transportation system is already available, although still expensive for full implementation. The approach taken by the US DOT is called Intelligent Transportation Systems (ITS). A major long-term element of ITS research and development effort is the Automated Highway System (AHS) which will provide fully automated "hands-off" operation at better levels of performance than current highway system in terms of safety, efficiency, and comfort. Current AHS research activities address the imminent user services involving safety advisory and driver assistance products such as in-vehicle warnings and emergency control intervention.

Vehicle control is probably the most important part of the advanced AHS applications. Implementation of AHS requires automatically controlled vehicles because technological requirements of the system are well beyond human capabilities. The past and present research on vehicle control emphasizes the importance of new methodologies in order to obtain efficient and safe longitudinal and lateral control. Combined lateral and longitudinal control needs to be implemented for full AHS applications. There is a large group of investigators working on vehicle control issues [Lasky93]. However, being able to control the dynamics of a vehicle does not necessarily mean that we have an AHS. "Intelligent" vehicle control methods will enable an automated vehicle to make decisions on its steering, and velocity commands, i.e., an automated vehicle will be able to plan its path. In an environment where there are multiple fast moving vehicles, making the right decision to avoid collisions and optimize the vehicle path is difficult.

Current concepts of intelligent vehicle control extend from independent vehicles to cooperative vehicles to infrastructure-supported systems [Shladover96]. At one end of the spectrum, autonomous vehicles with local sensing capabilities are envisioned; at the other end, fully hierarchical architectures with infrastructure-assisted vehicles are defined. The less autonomous the vehicle, the more the access the vehicle must have to the global environment information. The communications and control structures are complex in hierarchical approaches, while finding an optimal path will be a problem for independent vehicles. Initial research on automated vehicle control indicates that a planning system that can guarantee optimal operation with a sound theoretical background has not yet been developed, and it may be vital to AHS implementation [Lasky93].

We visualize two learning automata employing a reinforcement learning algorithm as the heart of our path planner. Using local sensor and limited communications data, the automata learn the optimal actions to be taken for a given situation. Sensor information is processed in virtual reinforcement modules that evaluate each action in the light of the current sensor status. Given enough time and correct learning parameters, the automata indicate the best actions to take and send these actions to the lower layer in the control hierarchy which in turn 'fires' the action.

The initial decision system uses mainly local information, and consequently, the actions learned by the intelligent controller are not globally optimal; the vehicles can survive, but may not be able to reach some of their goals. To overcome this problem, we treat pairs of automata as interconnected automata structures and visualize the interaction between vehicles as sequences of games played between automata. Every game corresponds to a "state" of the physical environment as described in Section 3. By evaluating these games, it is possible to design new decision rules, and to analyze the interactions by predicting the behavior and the positions of vehicles.

In the next section, we will briefly describe the intelligent control method using the learning and adaptation approach mentioned above. Section 3 describes the treatment of automata interactions as games, and the resulting analysis method. Example scenarios where we analyze the situations in the light of automata interactions follow in Section 4. A discussion of the results concludes the paper.

## **2. A Learning and Adaptive Control Method for Vehicle Navigation**

The first attempt to solve the problem of real-time decision making in a highway environment dates back to 1992. Mourou and Fade described “a planning method applicable to agents with perception and decision-making capabilities and the ability to communicate with other agents” [Mourou92]. Recent research on intelligent vehicle includes an adaptive intelligent vehicle module used in a simulation and design system for tactical driving algorithms [Sukthankar96]. Intelligent modules are designed to answer the need for real-time maneuver selection for short-term goals. This approach tries to find the suitable parameters for the modules that fire lateral and longitudinal actions. Another approach to intelligent control for autonomous navigation uses a decision-theoretic approach with probabilistic networks [Forbes95]. The problem is modeled as a partially observable Markov process, and the optimal action is a function of the current belief state described as joint distribution over all possible actual states of the world. It is similar to the work mentioned above in the sense of firing the actions. Similarly, a rule-based navigation system that uses worst-case decision making is defined by Niehaus and Stengel [Niehaus94]. Again, a stochastic model of the traffic situation based on sensor measurements is assumed.

Our approach differs from the above-mentioned works in the use of learning paradigm. Instead of learning the parameters affecting the firing of actions on repeated runs, the automata learn which action to fire based on the local sensor information. In other words, the learning is not in the design phase, but in the *run* phase. The learning parameters of the automata and the sensor modules define the capabilities of an automated vehicle, and can be used to model different behaviors. For example, large learning parameters decrease the decision time, and consequently, may result in a more “agile” vehicle path. Furthermore, there are no “prescribed conditions” for actions.

The idea of defining a “fixed” structure to be utilized to find the optimal action has its own appeal, since the performance of the system is deterministic in the sense that the best action for a specific situation is known. However, even a good driver does not follow rules deterministically. In this sense, the learning automata approach is able to capture the dynamics of driver behavior. The decision to “fire” an action is never taken at exactly the same time for similar conditions.

### **2.1. Stochastic Learning Automaton**

The concept of learning automata grew out of a fusion of the work in modeling observed behavior, the efforts to model the choice actions based on past observations, the attempts to implement optimal strategies in the context of the two-armed bandit problem, and the need for making rational decision in random environments [Narendra89]. The learning paradigm of the stochastic automaton is based on repeated actions and the resulting environment responses. One action is selected based on the action selection mechanism, the response (favorable or unfavorable) from the environment is observed, then the action selection mechanism is updated based on the response, and the procedure is repeated. The objective in the design of an automaton is to determine how the choice of the action at any stage should be guided by past actions and responses. The algorithm that guarantees the desired learning process is called a *reinforcement scheme*.

Classical and modern control techniques assume some level of knowledge of the system to be controlled, either in the form of the mathematical model of the process or the statistics of the uncertainties in the system. However, the assumptions on the system or its environment may be insufficient to successfully control the system if changes occur. It is then necessary to observe the process in operation, and obtain further knowledge of the system. Rule-based systems, although performing well on many problems, have the disadvantage of requiring complex modifications, even for minor changes in the problem space, and they cannot handle unanticipated situations. One approach is to view the control problem as a problem in learning. The idea behind designing a learning system is to guarantee robust behavior without the complete knowledge, if any, of the system to be controlled. A crucial advantage of reinforcement learning compared to other learning approaches is that it requires no information about the environment except for the reinforcement signal.

## 2.2 Learning Automata as Intelligent Vehicle Controller

For our model, we assume that an intelligent vehicle is capable of two sets actions. Lateral actions are *shift to left lane* (SL), *shift to right lane* (SR) and *stay in lane* (SiL). Longitudinal actions are *accelerate* (ACC), *decelerate* (DEC), and *keep the same speed* (SM). The actions SiL and SM are “idle actions,” and can be treated as a single action.

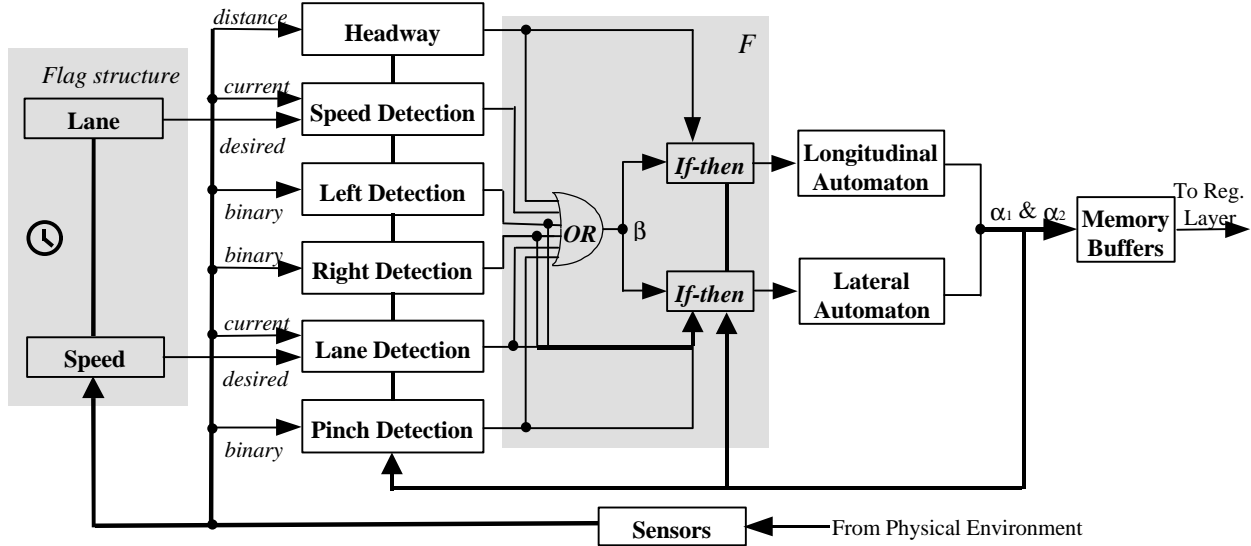
An autonomous vehicle must be able to ‘sense’ the environment around itself. In the simplest case, it is to be equipped with at least one sensor looking at the direction of the possible vehicle moves. Furthermore, an autonomous vehicle must have the knowledge of its own displacement. Therefore, we assume that there are four basic sensors on board the vehicle. These are the *headway sensor*, two *side sensors*, and a *speed sensor*. The headway sensor is a distance measuring device that returns the headway distance to the object in front of the vehicle. Side sensors are assumed to be able to detect the presence of a vehicle traveling in the immediate adjacent lane. The speed sensor is simply an encoder returning the current wheel speed of the vehicle.

Each sensor is connected to its associated decision module, which specifies an output signal in response to environmental data. In addition to these basic sensors, there may be two additional modules as shown in Figure 1. A *lane detection* sensor is assumed to return the current lane value, and a *pinch* module with local communication capabilities is able to detect the lateral actions of the neighboring vehicles. Sensor and communication modules evaluate the sensor signals in the light of current actions, and send a response to the automata. The response of the environment is a combination of the outputs of all four sensor modules (or “teacher blocks” for learning automata). The details of the mapping  $F$  are given in Section 2.2.2.

It is important to differentiate between the “automaton environment” and the “physical environment.” The output  $\alpha$  of an automaton is a signal that defines the current choice of action. It is the lower control layer’s (described as *regulation* layer in [Varaiya93]) responsibility to interpret this signal. When an action is carried out, it affects the physical environment (the vehicles and the highway). The sensors in turn sense the changes in this environment, and the feedback loop is closed with the sensor modules and the response signal  $\beta$ . The automata environment consists of the decision modules and the automata, and it is directly affected by the changes in the physical environment.

The regulation layer is not expected to carry out the chosen action immediately. To smooth the system output, only an action that is recommended  $m$  times consecutively by an automaton is carried out. In other words, the length of the *memory vector* is  $m$ . When this vector (or buffer) is filled with the same action, that action is fired. After an action is executed, the memory vectors

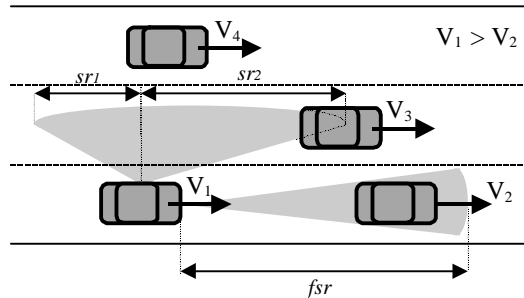
(lateral and longitudinal) are filled with idle actions (SiL or SM). A minimum processing speed of 25Hz is assumed, i.e., the action evaluation takes place at least 25 times a second. If the length  $m$  of the memory vectors is also 25, then there will be at least 1sec between consecutive actions. These are of course predefined parameters for the learning/control process.



**Figure 1.** Learning automata in a multi-teacher environment connected to the physical layers.

### 2.2.1 Sensor modules

The four basic sensor (or teacher) modules listed above are simple decision blocks that calculate the response associated with the corresponding sensor, based on the last chosen action. A penalty response (indicated by ‘1’) from the left side sensor is received only when the action is SL, and there is a vehicle in the left sensor’s range (defined by parameters  $sr_1$  and  $sr_2$ , Figure 2) or the vehicle is already traveling in the leftmost lane. Similarly, the action SR is penalized if and only if there is a neighboring vehicle in the right lane or the vehicle is already in the rightmost lane. All other situation-action combinations result in a reward response (‘0’) from these sensor modules. Actual side sensors may be able to only detect the presence of a vehicle in the adjacent lane, or there may be a sensor array capable of measuring the longitudinal distance to the neighboring vehicle. In the latter case, the information about the environment is more complex, and therefore requires more sophisticated decision rules for response signal evaluation [Ünsal97].



**Figure 2.** Sensor ranges for an autonomous vehicle.

Assuming that the front sensor is capable of detecting the headway distance, we can define the headway sensor module as shown in Table 1. The sensor range  $fsr$  describes the distance under which the presence of a vehicle is not desired. If the sensor “sees” a vehicle at a close distance, a

penalty response is sent to the automaton for actions SiL, ACC, and SM. All other actions (shifting lanes and decelerating) may serve to avoid a collision, and therefore, are encouraged. Again, the evaluation of the front sensor response based on the headway distance (and its rate of change) can be more complicated than the one given in Table 1, as described in [Ünsal97].

The speed sensor of the autonomous vehicle is assumed to be an encoder connected to the wheel shaft. The output of the encoder can be used to detect the wheel speed, which is approximately equal to vehicle speed (they are equal when cruising). The speed module's task is to compare the actual speed to the desired speed, and based on the action chosen, to send a feedback to the longitudinal automata. The decision rules depending on the speed difference and the current longitudinal action are given in Table 2. When the actual vehicle speed differs from the desired speed by more than a predefined amount, the action that will cause a decrease in the speed deviation receives a reward; others are penalized. This forces the vehicle to slow down or speed up in order to match the desired speed. The parameter  $pd$  defines the acceptable range of speed deviation.

**Table 1.** Output of the *Front Sensor* Module.

Action	Front Sensor Status	
	Vehicle in range	No vehicle in range
SiL	1	0
SL	0	0
SR	0	0
SM	1	0
ACC	1	0
DEC	0*	0

**Table 2.** Output of the *Speed Sensor* Module.

Action	Sensor Status		
	$dv < -pd$	$-pd < dv < pd$	$dv > pd$
SM	1	0	1
ACC	0	0	1
DEC	1	0	0

$dv \equiv \text{actual speed} - \text{desired speed}$   
 $pd \equiv \text{permitted difference between actual and desired speeds}$

The additional lane detection module is used to make path decisions more optimal as we describe later. A physical implementation of this module could be a vision system [Pomerlau96, Özgüner96]. Magnetic markers used for lateral control may provide position information. These solutions are still based on local sensing. Another alternative is to “feed” the present and desired lane information to the vehicles via broadcast communications. For our purposes, we will assume that an automated vehicle can sense its present lane, and that it has some idea about where it should be. Based on these two values, the action that leads to the necessary lane shift is encouraged by this teacher module.

It is imperative for an automated vehicle to make sure that the adjacent lane is not “claimed” by another vehicle before changing to that lane. A problem occurs when two vehicles one lane apart (e.g., vehicle 1 and 4 in Figure 2) shift to the same spot in the lane between them (“pinch condition”). In our simulations, we use the memory vector to check for other vehicles’ intentions to shift lanes. If the number of memory locations containing SR or SL is more than half the size of the vector for a vehicle, it is assumed that this vehicle is likely to shift lanes. If such an ‘intention’ signal is received

from a neighboring vehicle, the pinch module sends a penalty response for the action that may cause a problem. In a sense, the pinch module in an automated vehicle is driven by the memory vector of neighboring vehicles. In real implementation, this corresponds to a signaling vehicle which may be detected by a vision system, vehicle-to-vehicle communication indicating the intended actions, or a roadside-to-vehicle communication relaying the positions of vehicles.

Now that we have defined the sensor module outputs, the problem is to intelligently employ these signals for automata reinforcement. It is possible to treat all sensor modules as separate teachers with conflicting responses for the automata. For example, consider the situation given in Figure 2. Longitudinal action ACC will receive a penalty from the front sensor module due to the presence of the vehicle in front. If the actual speed of the autonomous vehicle is less than the desired speed, the speed module will try to force the vehicle to increase its speed. It is obvious that one of these conflicting feedback signals must have priority over the other. If a vehicle senses another vehicle occupying the immediate space in front of itself, it must slow down to avoid a collision no matter what its current speed is. The next section describes our approach to such a problem while introducing the learning mechanism.

The two flag structures shown in Figure 1 are defined in order to obtain a more optimal trajectory by temporarily altering the behavior of the vehicle. The *lane flag* enables the automated vehicle to take an action if it cannot reach its desired lane in a predefined time interval. This module keeps track of the time elapsed after a desired lane value is set (by on-board or roadside decision makers). If the vehicle cannot change to its desired lane in time, then the lane flag is set. The effect of this flag is to temporarily change the value of the desired speed. As a result, the vehicle slows down (or speeds up) in the hope of an opening to change lanes. Once the vehicle reaches its desired lane, the flag is reset, and the vehicle slowly changes its speed to match the previous desired speed.

Similar to the method described above, another flag to temporarily change the desired lane value is the *speed flag*. This module keeps track of the elapsed time after the current speed deviates from its desired value for the first time. If the vehicle is unable to adjust its speed in the predefined time interval, then the speed flag is set. This flag forces the lane detection module to send a penalty response to the lateral action SiL. Consequently, the vehicle changes lanes if there is an opening. Once the vehicle shifts lanes, the flag and the time counter are reset. The effectiveness of these two flags is demonstrated in [Ünsal97].

### 2.2.2 Learning Mechanism

Before describing the learning process for a stochastic automaton, we define the combined environment response. The outputs of the six teacher (i.e., sensor) modules described in the previous section are combined into a scalar response that can be used by the automaton for reinforcement purposes. As shown in Figure 1, the function  $F$  that maps multiple teacher responses into a single feedback signal<sup>2</sup> consists of an OR gate and two additional if-then condition blocks. These blocks provide the inhibition rules necessary for intelligent and safe navigation.

Table 3 shows the possible responses from the teacher modules for each action. Since a penalty response ('1') will inhibit a reward response ('0') by using an OR gate, the mapping is almost complete except for one problem with longitudinal action DEC. If the headway module returns a reward for this action, this must inhibit a penalty response from the speed sensor to guarantee safe operation. Furthermore, it is a good idea to override the penalty response to action SiL if the longitudinal action is DEC. This provides a smoother vehicle path, but requires additional range definitions for the front sensor, as described in [Ünsal97].

---

<sup>2</sup> Actually, there are two signals: one for lateral, another for longitudinal automata.

**Table 3.** Action-sensor module response matrix

Action	Modules					
	Headway	Left	Right	Speed	Lane	Pinch
SiL	0 / 1	0	0	0	0 / 1	0 / 1
SL	0	0 / 1	0	0	0 / 1	0 / 1
SR	0	0	0 / 1	0	0 / 1	0 / 1
SM	0 / 1	0	0	0 / 1	0	0
ACC	0 / 1	0	0	0 / 1	0	0
DEC	0 / 0*	0	0	0 / 1	0	0

Once we have the combined environment response for both automata, the control loop can be closed. As indicated before, a stochastic automaton learns from previous action and responses. A learning automaton generates a sequence of actions on the basis of its interaction with the environment. Each action  $a_i$  of the automaton is assigned a probability  $p_i$ ; the sum of all action probabilities is of course equal to 1. Since there is no basis in which the different actions  $a_i$  can be distinguished a priori, all action probabilities are initially equal ( $p_i = 1/r$  where  $r$  is the number of actions). After an action  $a_i$  is sent to the environment, the response  $b$  is observed. The action probabilities are then adjusted according to this response. If the response is favorable ('0'), then the probability  $p_i$  is increased; otherwise it is decreased. All other action probabilities  $p_j$  ( $j \neq i$ ) are adjusted accordingly in order to keep the sum of all probabilities equal to 1.

The algorithm used to update the action probabilities is called the reinforcement scheme. In general terms, a reinforcement scheme is a mapping  $T$  of the action probability vector, automaton action, and the environment response at time step  $n$  to the next action probability vector at time step  $n+1$ :

$$p(n+1) = T[p(n), a(n), b(n)]$$

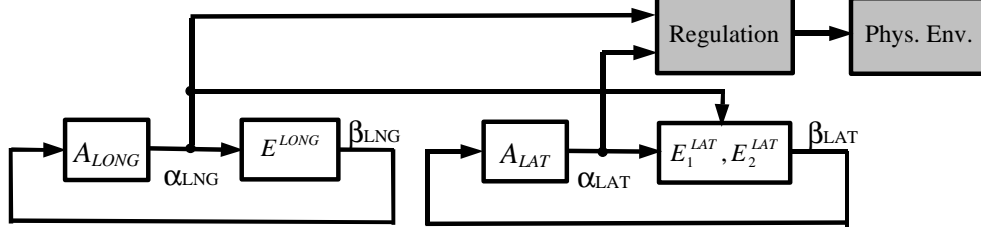
If  $p(n+1)$  is a linear function of  $p(n)$ , the reinforcement scheme is said to be linear; otherwise it is termed nonlinear. A variety of linear, nonlinear and hybrid schemes exists for stochastic automata [Narendra89]. The detailed description of the algorithms and simulation examples of the control method described here are given in [Ünsal97].

### 3. Learning Automata Games

In previous sections, we introduced an intelligent control method for autonomous navigation and path planning. The decision system mainly uses local information, and as a result, the actions learned by the intelligent controller are not globally optimal. The vehicles can survive, but may not be able to reach some of their navigational goals. To overcome this problem, we treat the interaction between vehicles as sequences of games played between pairs of automata. Every game corresponds to a "state" of the physical environment (the highway and the vehicles) as described below. By evaluating these games, it is possible to design new decision rules, and analyze the interactions by predicting the behavior and position of each vehicle.

The planning layer of an autonomous vehicle controller includes two automata, one each for lateral and longitudinal actions. The actions of each automaton, sooner or later, affects the physical environment, thus creating another level of interactions. Yet, there is a "direct interaction" between two automata in an intelligent vehicle due to the description of the teacher modules, and combination of the multiple teacher responses. Here, we will describe the interaction between automata in the same vehicle, and the automata in different vehicles. The nature of the non zero-sum games "played" is also defined.

Figure 3 shows the interaction structure between the two automata. As described before, both automata update their action probabilities based on the responses of the environment. Furthermore, the value of the current longitudinal action changes the environment response to the lateral automaton. In other words, the lateral environment<sup>3</sup> is determined by the current longitudinal action. The idea of interacting automata was first introduced in [Wheeler85]. The resulting configurations can be viewed as games of automata with particular payoff structures.



**Figure 3.** The longitudinal automaton determines the lateral automaton's environment (adapted from [Narendra89]).

We know that the lateral automaton  $A_{LAT}$  can operate in both environments  $E_i^{LAT}$  ( $i = 1, 2$ ). The difference between these two environments is the response of the headway module (inhibiting signal  $0^*$  in Table 3). In some situations, the choice of longitudinal action  $a_{LONG}$  affects the response of the lateral environment. All other environment changes are due to the changes in the physical environment, and we visualize these changes as state transitions as described below. Longitudinal automaton  $A_{LONG}$  is capable of converging to its best action using absolutely expedient schemes [Narendra89, Ünsal97]. The lateral automaton  $A_{LAT}$  in turn would converge to the best action in the environment determined by  $A_{LONG}$ .

Assume that the probabilities of receiving a penalty from the environment for all actions are known. For example, for an automated vehicle that finds itself in the rightmost lane of a two-lane highway after merging from an entry, the lateral action SR will receive penalty until the vehicle shifts lane. Consider the situation in the first few seconds where the automata environment is stationary. (With relatively fast update rates, this assumption is always possible.) Provided that the vehicle is in its desired lane and speed range, the environment response for the actions depends on the output of the headway and the left sensor module. Assume further that the probabilities of sensing a vehicle in front and side sensor ranges can be calculated for this particular case, based on the highway conditions. Then, by treating the probabilities of penalty as game payoffs for longitudinal and lateral automata, we can write the following game matrix of penalty probability pairs:

$$\begin{array}{c}
 \begin{array}{ccc}
 & SL & SR & SiL \\
 ACC & \left[ \left( \frac{7}{30}, \frac{4}{30} \right) \right. & \left. \left( \frac{7}{30}, 1 \right) \right. & \left. \left( \frac{7}{30}, \frac{5}{30} \right) \right] \\
 DEC & \left[ \left( 0, \frac{4}{30} \right) \right. & \left. (0, 1) \right. & \left. \left( 0, \frac{1}{30} \right) \right] \\
 SM & \left[ \left( \frac{4}{30}, \frac{4}{30} \right) \right. & \left. \left( \frac{4}{30}, 1 \right) \right. & \left. \left( \frac{4}{30}, \frac{5}{30} \right) \right]
 \end{array}
 \end{array}$$

First element of a payoff pair gives the probability of penalty for longitudinal action, while the second number is the penalty probability for the lateral action. Entries in the first and third rows correspond to first environment  $E_1^{LAT}$  for the lateral automaton, while the entries in the second row to the second environment  $E_2^{LAT}$ . The probability of penalty for lateral action SiL is less in the second environment where the longitudinal action is DEC. If the automata were not connected, an absolutely

<sup>3</sup> The word 'environment' here describes the automata environment, not the physical one.

expedient reinforcement scheme would force the automata to converge to actions DEC and SL (without the connection, lateral action SL is optimal since the penalty response from the front sensor is not suppressed). Based on this payoff (penalty probability) structure, this current solution action pair (DEC, SiL) is Pareto optimal and is an equilibrium point for this game.

Note that the situation above is very specific when we consider all sensors: the vehicles must be cruising at their desired speed range and lane, the pinch sensor must not send a penalty response, etc. For all other situations, the two automata may be considered ‘unconnected.’ Using the learning algorithms defined in the literature and described in [Ünsal97], the automata will converge to their best (optimal) actions separately. The interaction between automata is via the physical environment, which we consider to be stationary for the duration of a specific game. This results in a stationary automata environment. Of course, the solution of such a ‘disjoint game’ will be an equilibrium point (and a Pareto optimal solution) due to the convergence characteristics of the reinforcement schemes.

While the two automata in each vehicle form an interconnected pair that is guaranteed to reach the optimal solution for a stationary physical environment, interaction between vehicles creates another level of connection via the physical environment. The automata actions from other vehicles changes the physical environment which in turn affects the feedback responses sent to the automata (Figure 3). This type of interaction is indirect, and therefore cannot be formulated using a game matrix. Furthermore, the fact that such a game matrix will be time varying when considering multiple interacting vehicles complicates the matter.

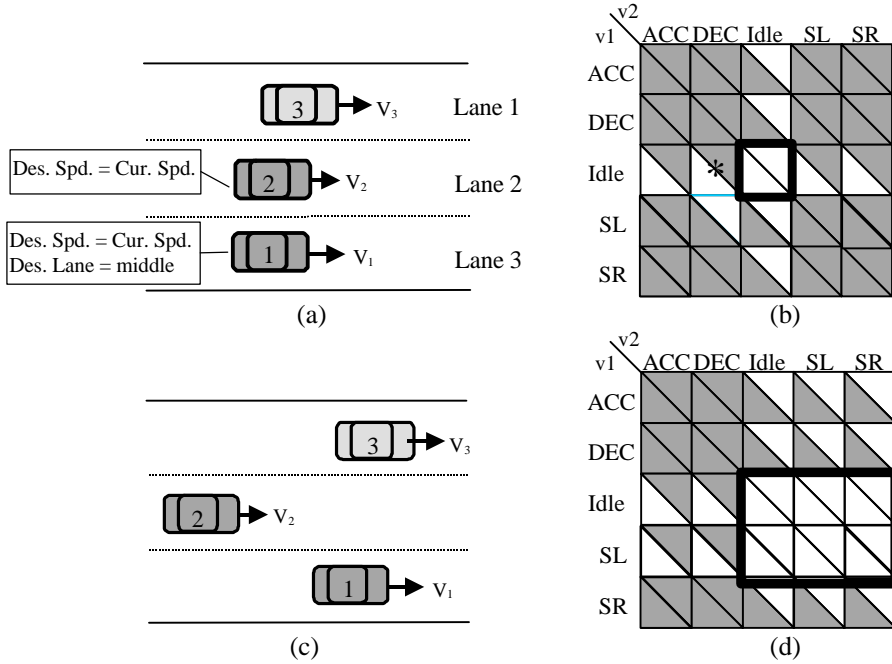
Instead, we treat the automata environments resulting from the ever-changing physical conditions as a switching environment. Based on certain changes in the physical world, the automata environment is changing from one state to another. Every state includes a different set of feedback responses to the lateral and longitudinal actions. All these different states of the environment are assumed to be stationary if the automata converge to the optimal actions long before another change takes place. As discussed previously, the automata are guaranteed to reach an intelligent decision in each stationary environment. Once a decision is made and sent to the regulation layer for maneuvers, corresponding actions are fired and the physical environment changes, forcing the automata environment to switch to another state.

It is important to realize that the actions need not be fired for the environment to switch from one state to another. For example, the physical environment may change due to speed differences between vehicles while only the idle actions (SiL and SM) are fired consecutively. The moment that one vehicle clears another vehicle’s sensor detection area, the automata environment changes. Similarly, when idle actions are fired, the physical and automata environments may not change. The interaction between the actions and the physical environment, and the physical and automata environment, are fairly complicated. Here, we will introduce a representation scheme that will facilitate the analysis of changes in the physical world in relation to the automata environment.

Illustrating vehicle interactions as automata games for every instance of the automata environment is not feasible, but it may be possible to define a similar matrix for all actions of autonomous vehicles. Consider a simple situation wherein two autonomous vehicles interact via their sensors, and communication (or signaling) devices. The physical presence of one vehicle affects the automata environment of the other. Note that the vehicles are not actually aware of the presence of other: the automata in each vehicle are simply trying to find the best action to take given a set of feedback responses.

Consider a situation with three vehicles as shown in Figure 4a. Vehicle 1 and 2 are autonomous; vehicle 3 is not automated. It is just an obstacle as far as the ‘intelligent’ vehicles are considered. Vehicle velocities are given as  $V_1 = V_3 > V_2$ . Vehicle 2 has no lane preference while

vehicle 1 needs to shift to middle lane. However, vehicle 1 cannot shift immediately to the middle lane because vehicle 2 is in its side sensor range. The automata environment for this situation is given in Figure 4b. (The actions SiL and SM are combined into a single action called *IDLE*. If a lateral action other than SiL is chosen, the row/column for combined action *IDLE* refers to the longitudinal idle action *SM*, and vice versa. If both SiL and *SM* are chosen, the table shows the OR-ed response.) Due to velocity differences, vehicle 2 drifts away from vehicle 1's sensor range (Figure 4c), and the automata environment switches to a new state (Figure 4d). In the mean time, the idle actions are fired repeatedly. With the new environment, the number of possible actions for vehicles 1 and 2 increases, and lateral action *SL* becomes the optimal solution for vehicle 1. Consequently, vehicle 1 changes lane which in turn causes another automata environment change.



**Figure 4.** Changes in the physical (l) and automata environments (r): vehicle 2 drifts away from vehicle 1's range. The matrices give the conditions in a particular automata environment resulting from physical location, current conditions, and predefined vehicle parameters. If the combined teacher response is a penalty, it is shown as a shaded triangle; reward responses are shown as white triangles. In Figure 4b and 4d, upper triangles are associated with vehicle 2, and optimal action pairs are indicated with black borders.

Using the same reasoning, we can establish which automata environment corresponds to each physical situation-vehicle condition pairs. The convergence to the optimal solution is guaranteed for all such situations, provided that the automata have enough time to learn. It is then possible to predict how the vehicle will react to a specific physical situation. This will enable us to define *highway scenarios* as described in the next section, and find solutions for intelligent path planning.

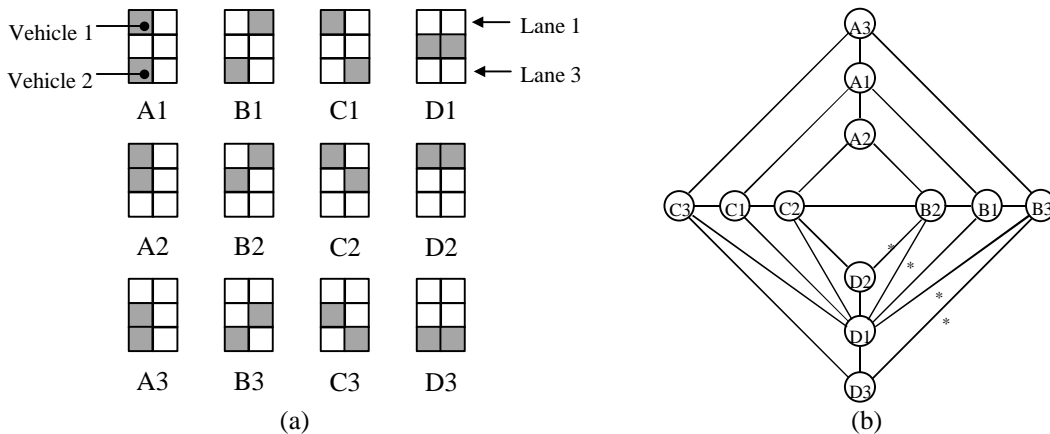
#### 4. Analysis of Highway Scenarios using State Transitions

Although vehicle controllers described in previous sections are able to avoid collisions, the resulting vehicle paths may not be the best solution for the problem of congestion. Furthermore, some of the vehicle paths may conflict and prevent the vehicles from reaching their desired goals. This problem with the autonomous vehicle approach has not yet been answered, while the infrastructure-supported systems inherently possess the methodology to solve the problem. We visualize a possible situation with multiple vehicles as a sequence of environment states. For all the states of the physical environment – which includes the positions of the vehicles and current parameters defining their

behavior – a corresponding automata environment can be defined. The automata environment is analyzed to predict possible (physical) environment changes. These changes are illustrated as state transitions. State diagrams formed using possible environment state transitions can be then used for analysis as well as design purposes.

Consider two vehicles sharing a 3-lane highway. All possible physical situations that arise while considering two vehicles in a three-lane highway are simplified to 12 states shown in Figure 5. Besides relative lateral positions, we assumed that only three possibilities exist for relative longitudinal positions. The distinguishing factor between these positions is whether a vehicle is in the side sensor range or not. There may be multiple corresponding automata environments to a given physical state due to several factors such as desired speed, desired lane, etc. The physical states can be represented by matrices in Figure 5. Each row in a matrix corresponds to a lane; each square indicates the presence of a vehicle in a road segment covered by side sensors. Not all possibilities are considered; instead, only the situations that are of interest for a specific scenario will be represented. Similar situations are also combined into a single state and simplified if necessary. Two situations are said to be similar if the sensor module outputs and/or possible actions are the same for both. Also note that for each state given in Figure 5, there is a reciprocal state with switched vehicle positions, denoted by an asterisk (e.g., B1\*); the actual number of states is 24).

To analyze the behavior of autonomous vehicles and the conflicts resulting from their interactions, we define highway scenarios. A highway scenario is a vector that combines physical location, sensor outputs, and internal parameters of vehicles. Once we know the automata environment at the beginning of a scenario, we can predict the (state) changes in the physical environment. Then, all possible changes are combined to form a state transition diagram showing the progression of the physical environment. The transitions between states are the direct results of the automata environment given by the matrices such as those given in Figure 4.



**Figure 5.** (a) Possible physical environment states for 2 vehicles in a 3-lane highway, and (b) state transition diagram for these states (self-transitions” are not shown; “\*” indicates a transition to a reciprocal state).

Now, consider the situation A1 in Figure 5 with two intelligent vehicles equipped with sensor modules except the flag structures discussed in Section 2.1.1. The velocities and lateral positions of the vehicles are the same. Suppose vehicle 1 needs to shift to lane 3, and vehicle 2 to lane 1. Since they are traveling at the same speed, there are no actions that would lead to a goal state using the basic sensor modules. Possible transitions are  $A1 \rightarrow A2$  and  $A1 \rightarrow A3$ . For transitions to these states, one of the vehicles must fill its memory vector with a lane shifting action. The transition must take longer if one of the vehicles (the one that is slow in filling its memory vector with lane shifting

action) detects the other vehicle signaling for lane shift. Eventually, this vehicle will give way to the other.

If the vehicles were to change speed, multiple transitions leading to goal states are possible. Suppose that vehicle 2 slows down. Then the automata environment shown in Figure 4b changes with the deletion of the penalty response for action DEC. The physical environment will then switch as a result of the longitudinal action taken. Therefore, the transitions given in Figure 6 are possible.

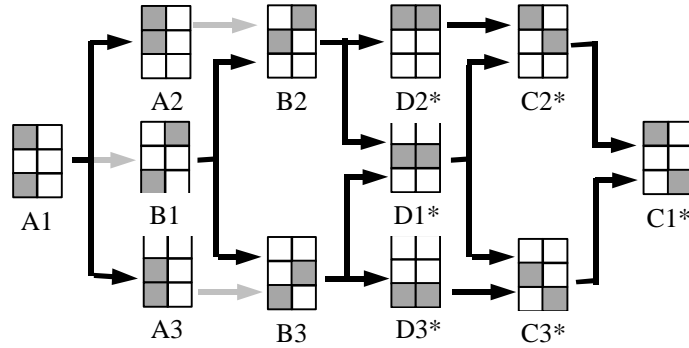


Figure 6. Possible transitions if vehicle 2 is able to slow down.

All transitions in Figure 6 except those indicated by gray color are automatic under the current circumstances. There are other possibilities solving the deadlock situations depending on the permitted longitudinal actions for vehicle 1 and 2. To introduce the change to the automata environment, the lane flag module described in Section 2.1.1 is designed. If a vehicle is unable to change to its desired lane in a predefined time interval, a flag that changes the vehicle’s desired speed (usually to a smaller value than the current speed) is set. Of course, the change must be different for required left and right shift to break the symmetry.

Consider a similar situation with three automated vehicles on a 3-lane highway (Figure 7a). As seen from the figure, the lateral positions and speeds of the vehicles are the same, and none of the vehicles is in its desired lane. Similar to the previous scenario, the solution lies in changing the relative speeds of the vehicles. Again, the lane flag is used to decrease the speeds of vehicles 1 and 2 to a smaller value than that of vehicle 3 (in order to break the symmetry). The state transition leading to a solution is similar to the one in previous example and is given in Figure 7b. A few other solutions are also possible if different speed adjustments are considered. There are 64 different situations for this scenario, only the ones that are encountered in this specific solution are shown.

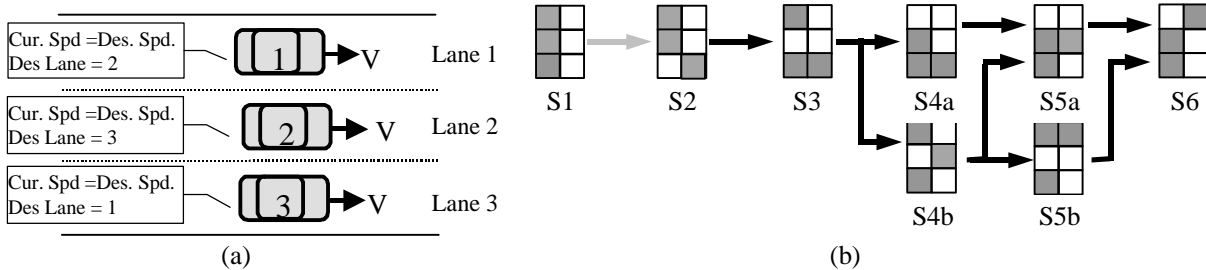
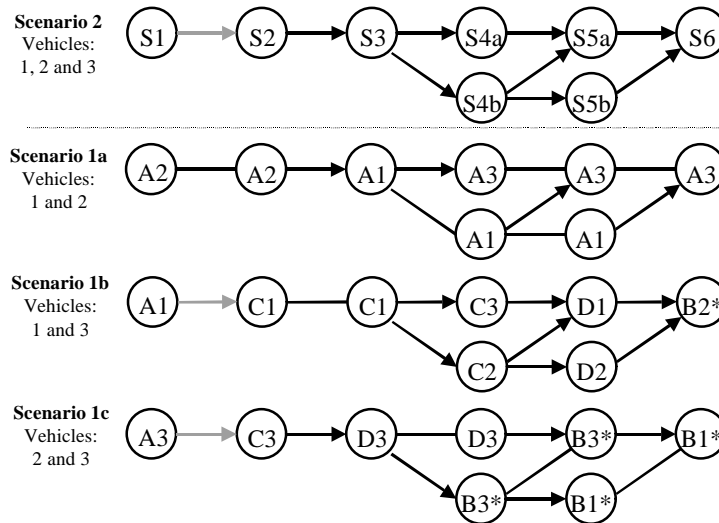


Figure 7. (a) Three vehicles with conflicting paths, and (b) a possible chain for this scenario.

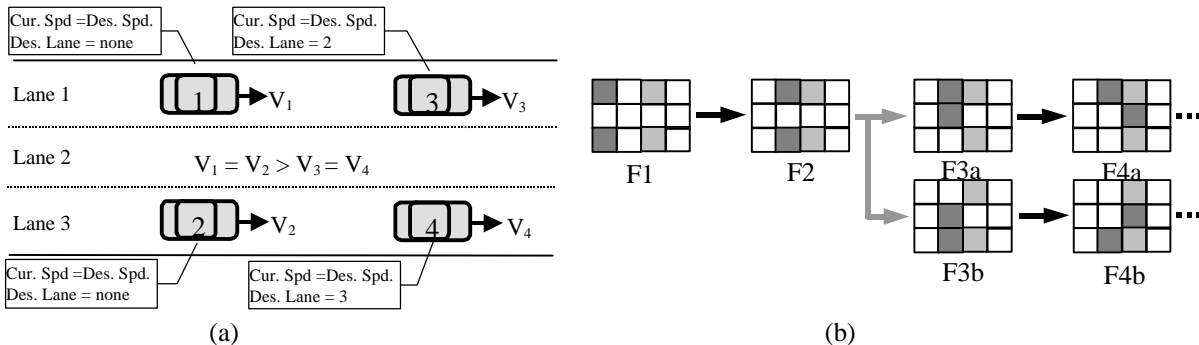
All transitions except the first one are automatic given current vehicle parameters. For the first transition, on the other hand, the lane flag needs to be set in at least in one vehicle (if it is vehicle 3). The problem and the solution for this case are similar to the two-vehicle scenario. This is not a coincidence; it is due to the *superposition* of the two two-vehicle situations. The term ‘superposition’ indicates that a given three-vehicle situation can be treated as three separate

asynchronous two-vehicle interactions. In terms of the two-vehicle states, the state transition diagram above can be written as separate transition diagrams using the two-vehicle situation previously described (Figure 8). The transitions that need to be forced by the lane flag are indicated with color gray, and they are (and must be) between corresponding states in both three-vehicle and two-vehicle transition diagrams. It is possible to define complex situations of multiple interacting vehicles as a group of many (conflicting and non-conflicting) two-vehicle situations. A complex scenario is nothing more than a superposition of multiple two-vehicle scenarios. The key transition that breaks the symmetry in multi-vehicle situation must correspond to at least one of the two-vehicle forced transitions.

Sometimes the interaction between two specific vehicles does not affect the multi-vehicle scenario considered for analysis. An example of this is given in Figure 9a. Vehicles 3 and 4 are traveling at their desired speeds and their desired lanes, unaware of the presence of the other vehicles. Assume that vehicles 1 and 2 are traveling at the same speed, which is faster than the others, while vehicle 3 and 4 are traveling at the same lateral position and speed. This situation is more complicated than previous examples. For vehicles 1 and 2 to keep their desired speed, it is necessary that they shift to the middle lane and pass vehicles 3 and 4. Although they are able to slow down to avoid collisions, the fact that they do not have a lane preference will prevent them from shifting to the middle lane. Since vehicles 3 and 4 do not interact with other vehicles, the situation becomes a two-vehicle situation with obstacles. Therefore, by analyzing the interactions between the first two vehicles, we must be able to find a solution to the problem.



**Figure 8.** Three-vehicle transition diagram can be written as three separate two-vehicle transition diagrams.



**Figure 9.** (a) Four vehicles with conflicting desired speeds, and (b) two possible solutions for the situation.

Since vehicles 1 and 2 are traveling faster than vehicles 3 and 4, the first transition to state F2 is automatic (Figure 9b). In state F2, two transitions that will solve the conflict, but need to be forced, are transitions to State F3a and F3b. Once one of these transitions takes place, the physical environment will move to the solution states. Since a desired lane value is not set for vehicles 1 and 2, another method is needed to force the environment to switch states. The solution to the problem is the speed flag we defined in Section 2.2.1. Under current circumstances, vehicles 1 and 2 will fire their idle actions repeatedly at state F2. With the addition of the speed flag, both vehicles will decide to shift to the middle lane after some time. Negotiation through the pinch modules will lead to state F3a or F3b. When the first vehicle that shifts to the middle lane clears the side sensor range of the second vehicle, the second vehicle will also shift to the middle and speed up.

## **5. Concluding Remarks**

Our non-model based approach to intelligent vehicle path control is based on two interacting stochastic learning automata. Instead of trying to foresee all possible traffic situations, we take the approach of defining a mechanism that can make intelligent decisions based on the local sensor information, keeping in mind the fact that initial phases of the AHS will include non-automated vehicles as well as intelligent vehicles capable of communicating with others. The method is capable of capturing the overall dynamics of the system that includes the vehicle, the driver, and the roadway. Definitions of the learning and sensor parameters determine the behavior of the each vehicle, and can be adjusted to guarantee safe operation.

Simulations of intelligent vehicles indicate the need for additional information sources other than local sensors. No matter what control structure is used for deployment, some form of communications between vehicles enhances the control. Although visual clues can be used to coordinate lateral actions, the lane changing capabilities of automated vehicles as well as the safety of the actions increase with local vehicle-to-vehicle communications. We have found that, in order to avoid pinch situations, vehicles may coordinate their lane changing actions by simply sending an 'intention' signal to neighboring vehicles.

Our attempt to design an intelligent path planner extends, to some degree, to other levels of vehicle control. For example, we have found that if a higher level of control/decision mechanism provides desired lane information, many local solutions (that are not globally optimal) may be extended to optimize overall traffic flow. There is a trade-off between what the automated vehicle can accomplish and how simple the sensing/information system is. The more global the information content of the decision mechanism, the more the vehicle can accomplish.

The method of evaluating possible environment state transitions based on associated automata environments enabled us to define additional decision mechanisms we called 'flags.' Speed and lane flags are used to solve the conflict situations arising from the multiple teacher responses and vehicle interactions. Although our method of evaluating the physical environment's state changes is based on the learning automata environment, similar methods can also be used with other decision mechanisms. By formal descriptions of the decision/control procedure, transition diagrams similar to those given in Section 4 can be created to analyze the highway situations.

## 6. References

- [Forbes95] Forbes, J., T. Huang, K. Kanazawa, and S. Russell, "The BATmobile: Towards a Bayesian Automated Taxi," *Proc. of the 14<sup>th</sup> Intl. Joint Conf. on Artificial Intelligence*, Montreal, Canada, 1995.
- [Lasky93] Lasky, T. L., and B. Ravani, "A review of Research Related to Automated Highway System (AHS)," Interim Rep. for FHWA, Contract no. DTFH61-93-C-00189, UC Davis, October 25, 1993.
- [Mourou92] Mourou, P., and B. Fade, "Multi-agent Planning and Execution Monitoring: Application to Highway Traffic," *Proc. of the AAAI Spring '92 Symposium*, pp. 107-112, 1992.
- [Narendra89] K. S. Narendra and M. L. Thathachar, *Learning Automata: An Introduction*, Englewood Cliffs, NJ: Prentice Hall, 1989.
- [Niehaus94] Niehaus, A., and R. F. Stengel, "Probability-Based Decision Making for Automated Highway Systems," *IEEE Trans. on Vehicular Tech.*, vol. 43, no. 3, pp. 626-634, 1994.
- [Özgüner96] Özgüner, Ü., M. Somerville, K. Redmill, C. Hatipoglu, K.A. Ünyelioglu, and D. Craig, "Experimental Results of a Lane Following Controller Based on a Vision Sensor," *Proc. of the 3<sup>rd</sup> World Congress in ITS*, Orlando, FL, October 1996.
- [Pomerleau96] Pomerleau, D., and T. Jochem, "Rapidly Adapting Machine Vision for Automated Vehicle Steering," *IEEE Expert*, vol. 11, no. 2, pp. 19-27, 1996.
- [Shladover96] Shladover, S. E., "Selection of Concepts for Automated Highway Systems (AHS)," *Proc. of the 3<sup>rd</sup> World Congress in ITS*, Orlando, FL, October 1996.
- [Sukthankar96] Sukthankar, R., J. Hancock, S. Baluja, D. Pomerleau, and C. Thorpe, "Adaptive Intelligent Vehicle Modules for Tactical Driving," *Proc. of the 13<sup>th</sup> National Conf. on Artificial Intelligence*, Portland, OR, 1996.
- [Ünsal97] Ünsal, C., "Intelligent Navigation of Autonomous Vehicles in an Automated Highway System: Learning Methods and interacting Vehicles Approach," Ph.D. Diss., Virginia Polytechnic Inst. & State Univ., Feb. 1997.
- [Varaiya93] Varaiya, P., "Smart Cars on Smart Roads: Problems of Control," *IEEE Trans. on Auto. Ctrl.*, vol. 38., no. 2, pp. 195-207, Feb. 1993.
- [Wheeler85] Wheeler, R. M. Jr., and K. S. Narendra, "Learning Models for Decentralized Decision Making," *Automatica*, vol. 21, pp. 479-484, 1985.